

PHS 675 Special Topics: Analysis of RNA-Sequencing Data

Homework 3: Read trimming

In this homework, we're going to use trimmomatic to trim all the raw reads in our sample directory. There are three new things to learn here:

1. Using variables and loops in Linux
2. Redirecting output from the console
3. Using a high-performance compute cluster to submit many jobs and wait for them to complete

The material at <https://bioinformatics.marshall.edu/training/Linux> might be useful for items 1 and 2 here (as well as more generally throughout the course). For item 1, read the pages "Variables in Bash" and "For loops in Bash". For item 2, read the page "Pipes and redirects" (for this homework, you really only need the sections from "Redirects" onwards).

This homework will be a different format to other homework in this course. In this homework, you will work through a scripted exercise on the server, and then answer questions about the code you used. When you submit the answers to the questions, I will also check your folder on the server to make sure the results look correct.

1. Log in to biocluster, activate the phs675 conda environment, and navigate to your folder on seqdata-new.

From there, change directories into phs675, data, and M-TH-20wk-Pancreas. Run

```
ls
```

If you have completed the work from class, you should see raw_fastqs, trimmed, and unpaired_trimmed.

2. Make a new directory called logs
3. Run the following. This will run trimmomatic on the second sample.

```
s="M-TH-Chow-20wk-2307-Pancreas"
```

```
trimmomatic PE \
```

```
raw_fastqs/${s}_R1.fastq.gz \
```

```
raw_fastqs/${s}_R2.fastq.gz \
```

```
trimmed/${s}_R1.fastq.gz \
```

```
unpaired_trimmed/${s}_R1.fastq.gz \
```

```
trimmed/${s}_R2.fastq.gz \
```

```
unpaired_trimmed/${s}_R2.fastq.gz \
```

```
ILLUMINACLIP:TruSeq3-PE-2-GGGGGG.fa:2:30:10 \
```

```
SLIDINGWINDOW:4:14 \
```

```
MINLEN:25 2>logs/${s}_trim.log
```

4. When it completes, list the contents of trimmed, unpaired_trimmed, and logs. Check the contents of the file in the logs folder (you can use cat to see the contents).

We are going to run trimmomatic on all the samples in the raw_fastqs folder. Since each one take a reasonable amount of time, we are going to use the fact we are working on a high-performance compute cluster, and submit each one of these to the cluster to run. The basic idea here is to do

```
srun [command] &
```

where [command] is the command we were previously using. srun here means "run on the cluster" and submits the job for running. The ampersand (&) at the end does this in the

background; this means it doesn't wait until the process is complete and lets us do something else in the console (such as submitting another job to the cluster).

Let's test this technique out on just one sample first. We can update the code from step 3 above by changing the variable to point to a new sample, and then modifying the call to use `srun`:

5. Run the following:

```
s="M-TH-Chow-20wk-2330-Pancreas"
srun -p normal trimmomatic PE \
raw_fastqs/${s}_R1.fastq.gz \
raw_fastqs/${s}_R2.fastq.gz \
trimmed/${s}_R1.fastq.gz \
unpaired_trimmed/${s}_R1.fastq.gz \
trimmed/${s}_R2.fastq.gz \
unpaired_trimmed/${s}_R2.fastq.gz \
ILLUMINACLIP:TruSeq3-PE-2-GGGGGG.fa:2:30:10 \
SLIDINGWINDOW:4:14 \
MINLEN:25 2>logs/${s}_trim.log &
```

(Don't omit the ampersand at the very end; also, do the next step reasonably soon after this one. Finally, the -p option to srun is specifying a "partition" of the cluster to use.)

6. Run the following

```
squeue
```

This will tell you everything that is running on the cluster. You should see your job there. When the job completes, you should get a message on the console. You can check the files are in the output folders and view the content of the log file as in step 4.

The next step is to write some code to run this on all samples. The first thing we will need is a list of all samples.

7. Try each of the following in turn, and inspect the output to figure out what they are doing. Use `man cut` and `man sort` to help understand these commands (man give the manual page for a command).

```
ls raw_fastqs
ls raw_fastqs | cut -d '_' -f 1
ls raw_fastqs | cut -d '_' -f 1 | sort -u
```

8. Run

```
samples=$(ls raw_fastqs | cut -d '_' -f 1 | sort -u)
```

Use

```
echo ${samples}
```

to see if the samples variable contains a list of all samples.

We are now in a position to run trimmomatic on each of our samples. I recommend reviewing the "For loops in Bash" page from the Linux tutorial linked above.

9. Clean up what you did before, by removing it (we're going to do it again, but better!):

```
rm trimmed/* unpaired_trimmed/* logs/*
```

10. Run the following:

```
for s in ${samples}; do
  srun -p normal trimmomatic PE \
    raw_fastqs/${s}_R1.fastq.gz \
    raw_fastqs/${s}_R2.fastq.gz \
    trimmed/${s}_R1.fastq.gz \
    unpaired_trimmed/${s}_R1.fastq.gz \
    trimmed/${s}_R2.fastq.gz \
    unpaired_trimmed/${s}_R2.fastq.gz \
    ILLUMINACLIP:TruSeq3-PE-2-GGGGGG.fa:2:30:10 \
    SLIDINGWINDOW:4:14 \
    MINLEN:25 2>logs/${s}_trim.log &
done
```

11. Check `squeue`. You should have 17 jobs submitted to the cluster. Take a break! You can log out if you want. Come back in an hour or so, and check the contents of your output folders (trimmed and unpaired_trimmed) and the log files. If you do a long listing of the trimmed files

```
ls -lh trimmed
```

you should see 34 files, ranging in size from about 70MB to about 320MB.

Questions and answers (write up for submission).

- I. Explain the code in step 3. What does each parameter to trimmomatic represent, and what do each of the parameters to the trimming steps represent. Refer to the trimmomatic manual at <http://www.usadellab.org/cms/?page=trimmomatic> if needed.
- II. Explain the first line of code in step 8. What does the code inside the parentheses do? What does the `$( ... )` construct mean? Refer to the "Pipes and Redirects" and "Variables in Bash" page from the Linux tutorial at <https://bioinformatics.marshall.edu/training/Linux>
- III. Explain the code in step 10; you need only explain the parts that are different to step 3.

- IV. Look at the log files. What is the range of percentages of reads for which both reads survived the trimming process? (For an advanced version of this, use `grep` to extract the important information.)